

Neuromorphic Computing for Embodied Intelligence in Autonomous Systems: Current Trends, Challenges, and Future Directions

Alberto Marchisio and Muhammad Shafique

eBRAIN Lab, Division of Engineering, New York University Abu Dhabi (NYUAD), Abu Dhabi, UAE
alberto.marchisio@nyu.edu, muhammad.shafique@nyu.edu

Abstract—The growing need for intelligent, adaptive, and energy-efficient autonomous systems across fields such as robotics, mobile agents (e.g., UAVs), and self-driving vehicles is driving interest in neuromorphic computing. By drawing inspiration from biological neural systems, neuromorphic approaches offer promising pathways to enhance the perception, decision-making, and responsiveness of autonomous platforms. This paper surveys recent progress in neuromorphic algorithms, specialized hardware, and cross-layer optimization strategies, with a focus on their deployment in real-world autonomous scenarios. Special attention is given to event-based dynamic vision sensors and their role in enabling fast, efficient perception. The discussion highlights new methods that improve energy efficiency, robustness, adaptability, and reliability through the integration of spiking neural networks into autonomous system architectures. We integrate perspectives from machine learning, robotics, neuroscience, and neuromorphic engineering to offer a comprehensive view of the state of the field. Finally, emerging trends and open challenges are explored, particularly in the areas of real-time decision-making, continual learning, and the development of secure, resilient autonomous systems.

Index Terms—Neuromorphic Computing, Embodied Intelligence, Autonomous Systems, Spiking Neural Networks.

I. INTRODUCTION

Autonomous systems, such as self-driving vehicles, drones, and wearable devices, require real-time decision-making under strict energy and computational constraints. Conventional architectures often struggle to meet these demands due to their high power consumption and latency. Neuromorphic computing, inspired by the energy efficiency of the human brain, offers a viable alternative through Spiking Neural Networks (SNNs), which process information via discrete events, enabling low-power, low-latency inference [1], [2]. This event-driven paradigm is particularly well-suited for handling temporally sparse data, aligning naturally with event-based sensing modalities.

Neuromorphic computing lies at the intersection of robotics, neuroscience, machine learning, and hardware design [3], [4]. However, its application to autonomous agents still requires cohesive strategies that address data encoding, model robustness, co-design of algorithms and hardware, and real-time adaptability. This paper presents a unified workflow that integrates recent advances across neuromorphic algorithms, hardware platforms, and event-based sensors, with cross-layer optimization techniques aimed at improving energy efficiency, security, reliability, and adaptability, thus laying a foundation for embodied intelligence in autonomous systems.

The key contributions of this paper are:

- A comprehensive workflow for developing secure, efficient, and reliable SNNs tailored for neuromorphic deployment in autonomous agents.
- Overview of design and optimization techniques of SNN architectures with emphasis on robustness, energy efficiency, and secure training through structural tuning, model compression, adversarial defenses, encryption techniques, and implementation of adaptive SNN mechanisms.
- Identification of current challenges and open research questions in neuromorphic systems, providing insights for future exploration.

II. NEUROMORPHIC COMPUTING WITH SNNs

A. Spiking Neural Networks

Spiking Neural Networks (SNNs), often considered the third generation of neural models, emulate biological neurons more closely than traditional networks [5]. Rather than processing continuous values, SNNs operate through discrete spikes: each neuron integrates incoming spikes into a membrane potential and fires when a threshold is exceeded (see Figure 1). This spike-based mechanism enables efficient processing of temporal patterns, making SNNs ideal for time-sensitive tasks such as motion recognition and real-time autonomous decision-making [4].

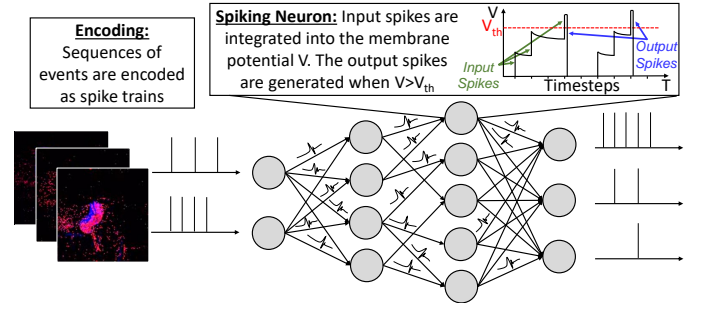


Fig. 1. Basic functionality of an SNN, where input events are encoded into spike trains, and the neuron integrates these into its membrane potential.

B. Neuromorphic Hardware

Neuromorphic platforms are specialized for executing SNNs efficiently by replicating structural and functional characteristics of biological systems. Examples include IBM's TrueNorth [6], Intel's Loihi [7], SpiNNaker, BrainScaleS [8], and commercial chips like Brainchip Akida and Synsense DYNAP-SE [9]. These systems are optimized for high computational throughput with minimal energy consumption, supporting scalable, low-power deployment of complex neural models in real-world environments.

III. CROSS-LAYER WORKFLOW FOR NEUROMORPHIC-BASED EMBODIED INTELLIGENCE IN AUTONOMOUS SYSTEMS

As shown in Figure 2, the proposed workflow begins with the acquisition of raw data through traditional or event-based sensors, followed by a pre-processing stage that includes event discretization and encoding. Yielding temporally structured event sequences suitable for spiking neural processing. Subsequently, a reliable SNN model is designed, incorporating hyperparameter tuning and fault tolerance for robustness against variability in inputs and system behavior. Afterward, a secure training pipeline combines adversarial training with encryption mechanisms to produce a resilient SNN. Next, hardware- and software-level optimizations, such as pruning, quantization, and hardware-aware NAS, are applied, guided by design constraints of the neuromorphic chip. The final step introduces real-time adaptation capabilities like continual learning and noise filtering, so the model can remain efficient under dynamic conditions. These stages result in a secure, efficient, and reliable SNN system that supports embodied intelligence in autonomous systems.

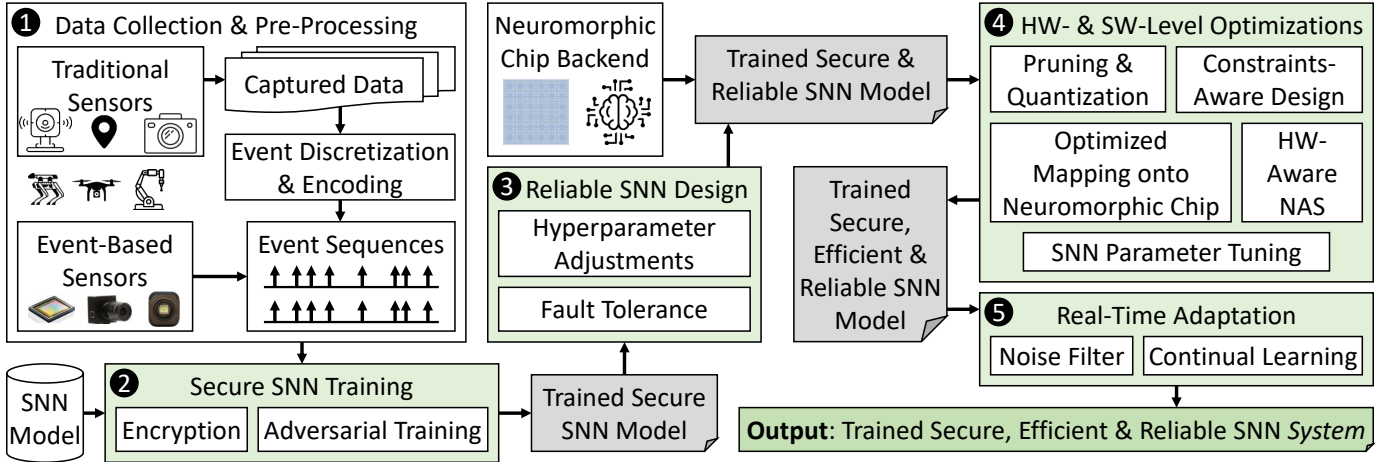


Fig. 2. Overview of the proposed workflow. After collecting data from sensors and pre-processing, secure SNN training is conducted to obtain a trained secure SNN model. Afterward, reliable SNN design is conducted to obtain a reliable SNN model on the neuromorphic chip backend. Further HW- and SW-level optimizations for energy efficiency are conducted. Real-time adaptation is integrated to obtain secure, efficient, and reliable SNN systems.

A. Data Collection and Pre-Processing

Data acquisition in neuromorphic systems often begins with event-based sensors, such as Dynamic Vision Sensors (DVS), that emit events when pixel-level brightness changes exceed a threshold [10]. This approach enables high temporal resolution and low-latency data capture, suitable for dynamic scenes. For scenarios where traditional frame-based sensors are employed, their data can be converted into event sequences through frame differencing or threshold-based encoding, mimicking sparse asynchronous events [11]. Pre-processing these streams requires optimizing for efficient SNN representation. Techniques such as noise filtering, event denoising, and temporal smoothing help reduce redundancy and improve signal quality. Additionally, methods like time-surface generation [12] and histogram-based representations [13] help capture spatiotemporal patterns for better SNN learning and inference.

B. Secure SNN Training

Adversarial Training of SNNs: Securing SNNs is critical for safe, privacy-aware deployment. A key aspect is robustness against adversarial attacks. Techniques such as adversarial training inject crafted perturbations during training to help the model learn features resilient to attacks [14]. Recent work [15] introduces spike-aware regularization tailored to the temporal dynamics of SNNs, boosting robustness without major accuracy loss.

Encrypted SNNs: Encryption further protects sensitive data during inference. Homomorphic encryption schemes like the Brakerski/Fan-Vercauteren (BFV) [16] enable encrypted spike processing without revealing raw data. This secures user data even in untrusted environments, though at added cost. As secure neuromorphic computing evolves, combining adversarial robustness and encryption is becoming essential for trustworthy deployment.

As shown in Figure 3, the Spiking-AlexNet maintains competitive accuracy under encryption compared to the conventional AlexNet model. It also shows improved resilience in low-modulus regimes where quantization effects dominate, suggesting better alignment of SNNs with encrypted computation constraints [17].

C. Reliable SNN Design

SNN Hyperparameter Adjustments: SNNs possess unique structural characteristics that are tunable for robustness. A recent study highlights how structural parameters like voltage threshold V_{th} and spike integration window T affect model resilience [18]. Adjusting these parameters changes spiking dynamics, affecting

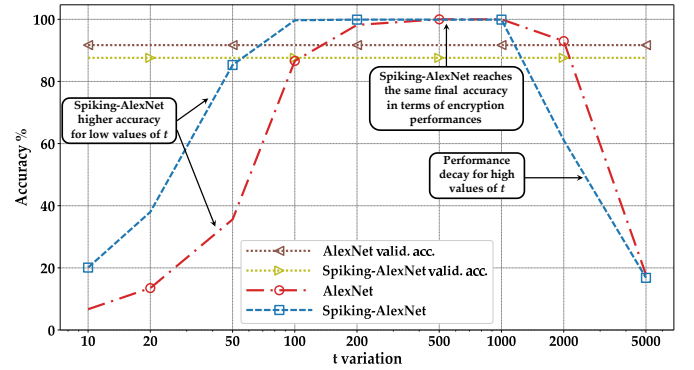


Fig. 3. Accuracy comparison between AlexNet and Spiking-AlexNet for the FashionMNIST dataset, when the models are encrypted using the BFV scheme, varying the plaintext modulus t . Figure adapted from [17].

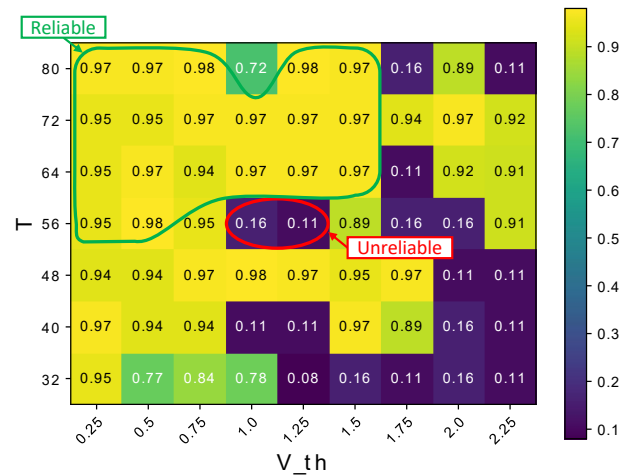


Fig. 4. Robustness comparison of 4-layer SNNs on MNIST, varying the voltage threshold V_{th} and time window T . Figure adapted from [18].

susceptibility to perturbations. Lower thresholds increase firing and robustness at the cost of energy, while longer time windows improve stability. Figure 4 shows how proper tuning boosts robustness on MNIST with minimal architectural changes.

SNN Fault Tolerance: Beyond hyperparameters, model reliability under faults is crucial, especially on embedded neuromorphic

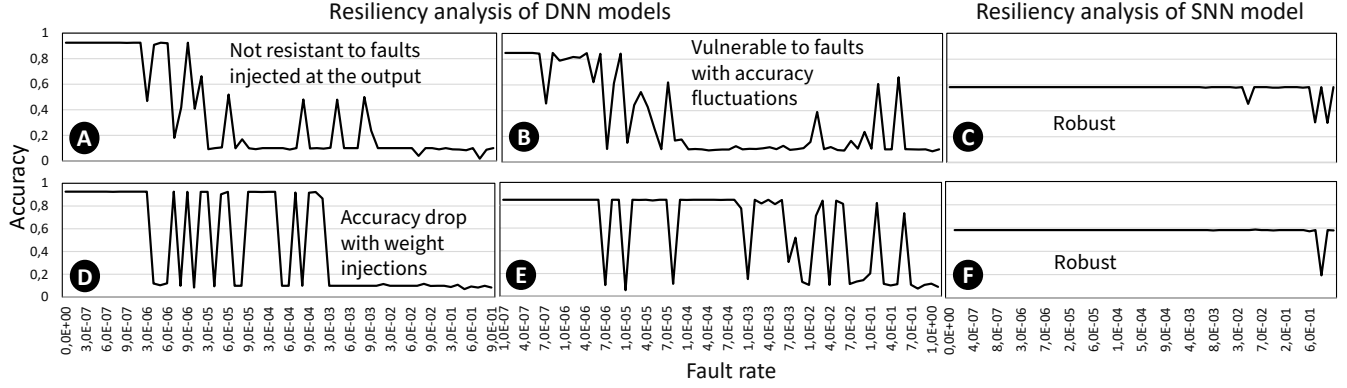


Fig. 5. Comparison of test accuracy for different models under fault injection. A: Output injection on the ResNet18 for the CIFAR-10 dataset. B: Output injection on the VGG11 for the CIFAR-10 dataset. C: Output injection on a 4-layer SNN on the DVS-Gesture dataset. D: Weight injection on the ResNet18 for the CIFAR-10 dataset. E: Weight injection on the VGG11 for the CIFAR-10 dataset. F: Weight injection on a 4-layer SNN on the DVS-Gesture dataset. Figure adapted from [19].

systems. The *enpheeph* framework [19] analyzes SNN robustness via fault injection. As seen in Figure 5, SNNs outperform ResNet18 and VGG11 in maintaining accuracy under both output and weight faults on CIFAR-10 and DVS-Gesture. This highlights the inherent fault tolerance of SNNs due to their sparse and distributed activation.

D. HW- & SW-Level Optimizations for SNNs

SNN Compression: Compression techniques such as quantization and pruning are crucial for reducing energy and memory demands of SNNs, especially in edge devices. Reducing weight count and precision reduces energy use with minimal performance loss. The SNN4Agents framework [20] shows that a 10-bit CarSNN model preserves high accuracy on NCARS, with accuracy only dropping significantly below 8 bits, highlighting the trade-off between bitwidth and performance.

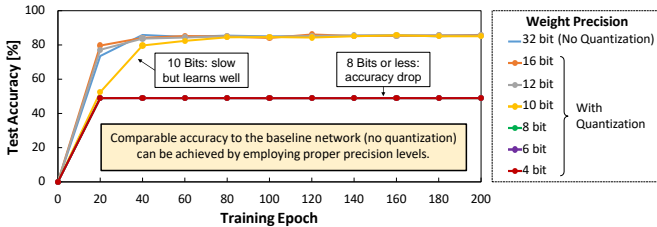


Fig. 6. Accuracy of the CarSNN model on the NCARS dataset across different precision levels (from 32 bits to 4 bits). Figure adapted from [20].

Optimized SNN Learning Rate Policies: Training efficiency is key for sustainable SNN development. FastSpiker [21] leverages learning rate strategies such as exponential decay and cosine warm restarts for faster training without sacrificing accuracy. As shown in Figure 7, these methods offer strong gains in efficiency, notably reducing carbon emissions.

HW-Aware NAS and Optimized Parameter Tuning: Optimizing SNNs through hardware-aware NAS allows for performance-driven architecture search under memory and energy constraints [22]. In [20], varying precision and attention window size in CarSNN yields up to 97.32% memory savings (see Figure 8). Even 16-bit configurations achieve significant compression without much performance loss, enabling scalable SNN deployment in constrained systems.

E. Real-Time Adaptation of SNNs

SNN Noise Filters: In dynamic environments, event-based sensors frequently produce noisy spikes that degrade SNN performance.

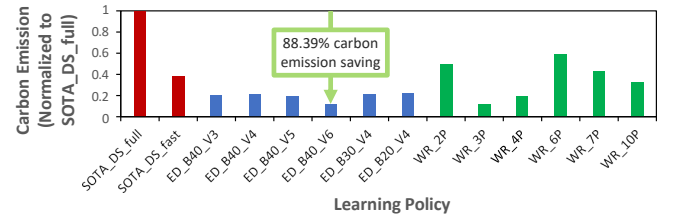


Fig. 7. Exploration of different learning rate policies (exponential decay with different values of batch size and voltage threshold, warm restarts with different numbers of peaks) for the CarSNN model on the NCARS dataset and their impact on the carbon emission, compared to the baseline. Figure adapted from [21].

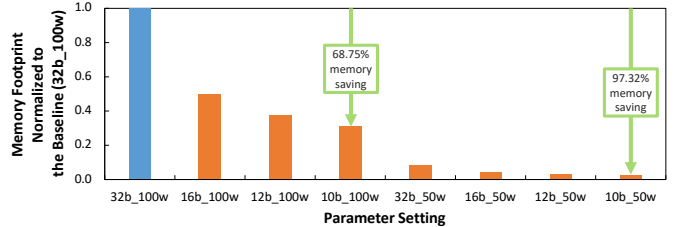


Fig. 8. Exploration of different parameter configurations (number of bits and size of the attention window) of the CarSNN model on the NCARS dataset and their impact on the SNN memory footprint, compared to the baseline. Figure adapted from [20].

Filtering techniques like background activity filters help suppress irrelevant input spikes. R-SNN [23] integrates such filters to improve robustness in noisy or adversarial settings by attenuating noise before it reaches spike integration.

Continual Learning for SNNs: Continual learning enables a model to incorporate new knowledge without forgetting old tasks. lpSpikeCon [24] supports this via low-precision synaptic weights and online updates, enabling multi-task learning with reduced memory needs. As shown in Figure 9, a 200-neuron SNN trained with lpSpikeCon retains significantly higher average accuracy, especially early ones, even under 14-bit quantization.

IV. CHALLENGES AND OPEN RESEARCH QUESTIONS OF NEUROMORPHIC SYSTEMS

Despite substantial progress in deploying SNNs on neuromorphic hardware, several key challenges and open research questions persist. Tackling these issues is vital for broader adoption and effective real-world deployment.

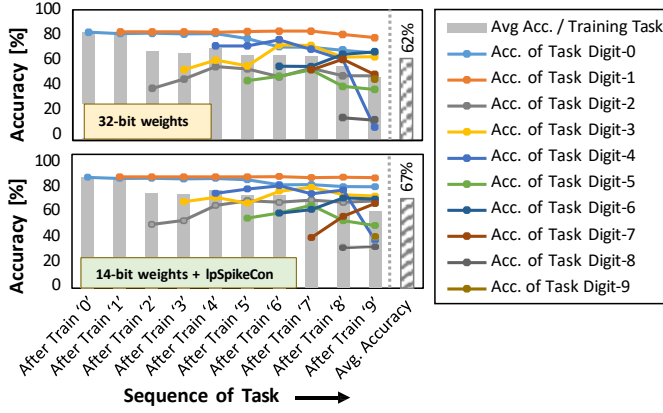


Fig. 9. Accuracy of a 200-neuron SNN for the MNIST dataset, trained on different tasks, while also reducing the bitwidth. Figure adapted from [24].

- **Hardware Implementation and Integration:** Although SNNs enable energy-efficient processing, deploying them on neuromorphic hardware is constrained by limited precision and scalability. Bridging the algorithm-hardware gap demands co-design strategies that align model architecture with hardware capabilities [25].
- **Security and Robustness Against Backdoor Attacks:** SNNs remain vulnerable to backdoor attacks, where adversarial patterns are introduced during training. Ensuring integrity through detection and defense mechanisms remains an active area for secure deployment [26].
- **Standardization and Benchmarking:** The absence of consistent benchmarks and evaluation protocols impedes fair comparison across neuromorphic platforms. Community-driven standard frameworks are needed to support reproducibility and accelerate progress [27].
- **Scalability of Neuromorphic Hardware:** Current neuromorphic systems face limitations in scaling to support complex, real-world workloads. Addressing challenges related to communication overhead, power efficiency, and thermal constraints is crucial for their widespread use [28].

V. CONCLUSION AND ROAD AHEAD

This study outlines an end-to-end methodology for constructing secure, high-performance, and dependable SNNs optimized for neuromorphic applications in autonomous systems. The process begins with the acquisition and preprocessing of sensory data, including both traditional and event-based modalities. Subsequently, the development of robust SNN architectures is addressed through hyperparameter tuning and fault tolerance. To fortify these networks against potential threats, the integration of security measures, including adversarial robustness techniques and encryption-based safeguards, is explored. To meet stringent energy and memory limitations, the workflow incorporates model compression techniques, adaptive learning rate schedules, and architectural exploration. Real-time adaptability is addressed through noise filtering and continual learning mechanisms to ensure robustness in changing environments. Looking ahead, the advancement of neuromorphic computing for embodied intelligence will depend on scalable co-design of SNN models and hardware, efficient deployment toolchains, and standardized benchmarking. As the domain progresses, the fusion of bi-inspired neural architectures with hardware acceleration is poised to unlock a new generation of intelligent, adaptive, and energy-aware autonomous systems.

ACKNOWLEDGMENT

This work was supported in parts by the NYUAD Center for Cyber Security (CCS), funded by Tamkeen under the NYUAD Research Institute Award G1104, and the NYUAD Center for Artificial Intelligence and Robotics (CAIR), funded by Tamkeen under the NYUAD Research Institute Award CG010.

REFERENCES

- [1] A. Viale *et al.*, “Carsnn: An efficient spiking neural network for event-based autonomous cars on the loihi neuromorphic research processor,” in *IJCNN*, 2021.
- [2] A. Viale *et al.*, “Lanesnns: Spiking neural networks for lane detection on the loihi neuromorphic processor,” in *IROS*, 2022.
- [3] A. Marchisio and M. Shafique, “Embedded neuromorphic using intel’s loihi processor,” in *Embedded Machine Learning for Cyber-Physical, IoT, and Edge Computing: Software Optimizations and Hardware/Software Codesign*, Springer, 2023.
- [4] R. V. W. Putra *et al.*, “Embodied neuromorphic artificial intelligence for robotics: Perspectives, challenges, and research development stack,” in *ICARCV*, 2024.
- [5] W. Maass, “Networks of spiking neurons: the third generation of neural network models,” *Neural networks*, 1997.
- [6] F. Akopyan *et al.*, “Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip,” *IEEE TCAD*, 2015.
- [7] G. Orchard *et al.*, “Efficient neuromorphic signal processing with loihi 2,” in *SiPS*, 2021.
- [8] C. Pehle *et al.*, “The brainscales-2 accelerated neuromorphic system with hybrid plasticity,” *Frontiers in Neuroscience*, 2022.
- [9] O. Richter *et al.*, “Dynap-se2: a scalable multi-core dynamic neuromorphic asynchronous spiking neural network processor,” *Neuromorphic Computing and Engineering*, 2024.
- [10] P. Lichtsteiner, C. Posch, and T. Delbruck, “A 128x128 120 db 15us latency asynchronous temporal contrast vision sensor,” *IEEE JSSC*, 2008.
- [11] A. Amir *et al.*, “A low power, fully event-based gesture recognition system,” in *CVPR*, 2017.
- [12] X. Lagorce *et al.*, “Hots: A hierarchy of event-based time-surfaces for pattern recognition,” *IEEE TPAMI*, 2016.
- [13] A. Sironi *et al.*, “Hats: Histograms of averaged time surfaces for robust event-based object classification,” in *CVPR*, 2018.
- [14] A. Madry *et al.*, “Towards deep learning models resistant to adversarial attacks,” in *ICLR*, 2018.
- [15] J. Ding *et al.*, “Snn-rat: Robustness-enhanced spiking neural network through regularized adversarial training,” *NeurIPS*, 2022.
- [16] J. Fan and F. Vercauteren, “Somewhat practical fully homomorphic encryption,” *Cryptology ePrint Archive*, 2012.
- [17] F. Nikfam *et al.*, “A homomorphic encryption framework for privacy-preserving spiking neural networks,” *Information*, 2023.
- [18] R. El-Allami *et al.*, “Securing deep spiking neural networks against adversarial attacks through inherent structural parameters,” in *DATE*, 2021.
- [19] A. Colucci, A. Steininger, and M. Shafique, “enpheap: A fault injection framework for spiking and compressed deep neural networks,” in *IROS*, 2022.
- [20] R. V. W. Putra, A. Marchisio, and M. Shafique, “Snn4agents: a framework for developing energy-efficient embodied spiking neural networks for autonomous agents,” *Frontiers in Robotics and AI*, 2024.
- [21] I. Bano *et al.*, “Fastspiker: Enabling fast training for spiking neural networks on event-based data through learning rate enhancements for autonomous embedded systems,” in *ICARCV*, 2024.
- [22] I. Bano *et al.*, “A methodology to study the impact of spiking neural network parameters considering event-based automotive data,” in *ICARCV*, 2024.
- [23] A. Marchisio *et al.*, “R-snn: An analysis and design methodology for robustifying spiking neural networks against adversarial attacks through noise filters for dynamic vision sensors,” in *IROS*, 2021.
- [24] R. V. W. Putra and M. Shafique, “Ipspikecon: Enabling low-precision spiking neural network processing for efficient unsupervised continual learning on autonomous agents,” in *IJCNN*, 2022.
- [25] P. K. Huynh *et al.*, “Implementing spiking neural networks on neuromorphic architectures: A review,” *arXiv preprint arXiv:2202.08897*, 2022.
- [26] L. Jin, M. Lin, W. Jiang, and J. Zhan, “Data poisoning-based backdoor attack framework against supervised learning rules of spiking neural networks,” *arXiv preprint arXiv:2409.15670*, 2024.
- [27] D. R. Muir and S. Sheik, “The road to commercial success for neuromorphic technologies,” *Nature Communications*, 2025.
- [28] D. Kudithipudi *et al.*, “Neuromorphic computing at scale,” *Nature*, 2025.